

UE- SURVEILLANCE & SECURITE DES RESEAUX

Dr Lambert KADJO

lamberttanon_k@yahoo.fr

<http://www.kadjo-lambert.c4.fr>

UE- SURVEILLANCE & SECURITE DES RESEAUX

Chapitre 1: GENERALITES SUR LA SECURITE DES RESEAUX

- 1.1 MENACES ET ATTAQUES
- 1.2 INTERROGATIONS ET SENSIBILISATION À LA SÉCURITÉ
- 1.3 SERVICES DE LA SECURITE DES RESEAUX

Chapitre 2: PRINCIPES DE CRYPTOGRAPHIE

- 2.1 CONCEPTS DE BASE
- 2.2 ALGORITHMES CRYPTOGRAPHIQUES
- 2.3 FONCTIONS DE HACHAGE

Chapitre 3: TYPOLOGIE DES SOLUTIONS DE SECURITE DES DONNEES NUMERIQUES

- 3.1 CERTIFICATS NUMERIQUES, INFRASTRUCTURES A CLES PUBLIQUES
- 3.2 PROTOCOLES SSL/TLS
- 3.3 AUTHENTIFICATION UNIQUE: SINGLE SIGN-ON

Chapitre 4: TYPOLOGIE DES SOLUTIONS DE SECURITE ET DE SURVEILLANCE DES RESEAUX

- 4.1 DURCISSEMENT DES EQUIPEMENTS RESEAUX
- 4.2 PROTOCOLE SSH
- 4.3 PARE-FEUX
- 4.4 RESEAUX VIRTUELS: VLAN & VPN
- 4.5 SYSTEMES DE DETECTION/PREVENTION D'INTRUSION/ PRÉVENTION DES PERTES DE DONNÉES (DLP Data Loss Prevention)
- 4.6 SECURITE DES RESEAUX WI-FI
- 4.7 PROTOCOLE SNMP ET OUTILS DE SUPERVISION

Chapitre 5: POLITIQUES ET MODELES DE SÉCURITÉ

- 5.1 CRITERES D'EVALUATION
- 5.2 MODELES POUR LA POLITIQUE DE SECURITE
- 5.3 REFERENTIELS ET NORMES DE BONNES PRATIQUES POUR LA SECURITE DES RESEAUX

UE- SURVEILLANCE & SECURITE DES RESEAUX

Chapitre 2: PRINCIPES DE CRYPTOGRAPHIE

2.1 CONCEPTS DE BASE

2.2 ALGORITHMES CRYPTOGRAPHIQUES

2.3 FONCTIONS DE HACHAGE

Vocabulaire associé à la cryptographie

- ❖ **Cryptologie** : Science mathématique comportant deux branches : la cryptographie et la cryptanalyse.
- ❖ **Cryptographie** : Etude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné.
- ❖ **Chiffrement** : Méthode cryptographique consistant à transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction inverse permettant de retrouver le texte clair à partir du texte chiffré est appelée **Déchiffrement**.
- ❖ **Texte clair (Clear text ou Plain text)** : Caractères ou bits sous une forme lisible par un humain ou une machine.
- ❖ **Texte chiffré (Cipher text) ou cryptogramme**: Résultat de l'application d'un chiffrement à un texte clair.
- ❖ **Clé** : paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un **algorithme symétrique**, la clé est identique lors des deux opérations. Alors que pour les algorithmes asymétriques, la clé de chiffrement diffère de celle de déchiffrement.
- ❖ **Cryptanalyse** : Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.
- ❖ **Cryptosystème** : défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné.
- ❖ **Décryptage**: action permettant de retrouver le texte clair sans connaître la clé de déchiffrement.

Vocabulaire associé à la cryptographie

❖ **Codage par flux** : Un générateur pseudo aléatoire génère une suite de bits qui est combinée bit à bit avec le texte clair pour former le cryptogramme. La même suite aléatoire doit être utilisée pour reconstituer le clair. Comme on utilise l'opération XOR (ou exclusif) pour combiner les bits, les opérations de chiffrement et déchiffrement sont identiques.

❖ **Codage par blocs**: Ce sont en fait des blocs de données du texte en clair qui sont en réalité chiffrés (ou déchiffrés), généralement de 64 bits. Exemples: méthode ECB (Electronic Code Block), méthode CBC (Cipher Bloc Chaining), méthode CFB (Cipher Feed Back), méthode OFB (Output FeedBack).

✓ Plus robustes aux attaques.

Services de sécurité associés à la cryptographie

- ❑ Le but de la cryptographie traditionnelle est d'élaborer des méthodes permettant de transmettre des données de manière confidentielle par chiffrement;
- ❑ Le but de la cryptographie moderne est de traiter plus généralement des problèmes de sécurité des communications et de fournir un certain nombre de services de sécurité:
 - ❖ **Intégrité des données** : Le contrôle d'intégrité d'une donnée consiste à vérifier que cette donnée n'a pas été altérée, frauduleusement ou accidentellement.
 - ❖ **Contrôle d'accès** : Il s'agit d'authentifier les utilisateurs de façon à limiter l'accès aux données, serveurs et ressources aux seules personnes autorisées (un mot de passe pour un disque dur, par exemple).
 - ❖ **Confidentialité** : Il s'agit de rendre l'information inintelligible à tous les opposants tant lors de sa conservation qu'au cours de son transfert par un canal de communication.
 - ❖ **Authentification**: consiste à s'assurer que le destinataire est bien celui qui prêtant être (authentification des partenaires) et d'obtenir une garantie que l'expéditeur a bien signé l'acte (authentification de l'origine des informations).
 - ❖ **Non répudiation**: Il s'agit de garantir l'authenticité de l'acte. L'expéditeur ne peut nier le dépôt d'information, le réceptionneur ne peut nier la remise d'information, ni l'un ni l'autre ne peut nier le contenu de cette information.
 - ❖ **Anti-rejeu**. Un attaquant obtient une copie d'un message valide et la transmet ultérieurement à la destination

Principes qui régissent la sécurité des systèmes modernes

Les cryptosystèmes modernes ont leur sécurité exprimée en ces termes :

« À moins que l'adversaire ne passe un temps déraisonnable à calculer, le système X est sûr. »

En fait, la sécurité de ces systèmes est habituellement exprimée en des termes plus faibles :

« En supposant que le problème Y ne peut être résolu en temps raisonnable, le système X est sûr. »

Techniques de base

- ❑ Les mécanismes de sécurité sont basés sur un ensemble d'outils cryptographiques :
 - Les algorithmes de chiffrement : **Symétrique ou à clé secrète, Asymétriques ou à clé publique**. Ici, la propriété de base est que pour un message M en clair, si
 - ✓ C est le texte chiffré,
 - ✓ K est la clé (dans le cas d'un algorithme à clé symétrique), K_E (resp. K_D) est la clé de chiffrement (resp. la clé de déchiffrement) dans le cas d'algorithmes asymétriques,
 - ✓ $E(x, y)$ est la fonction de chiffrement avec x , le message et y , la clé,
 - ✓ $D(x, y)$ est la fonction de déchiffrement avec x , le message et y , la clé
- on a
$$M = D(C, K) = D(E(M, K), K) \text{ pour le cas symétrique où } C = E(M, K)$$
$$M = D(C, K_D) = D(E(M, K_E), K_D) \text{ pour le cas asymétrique où } C = E(M, K_E)$$
- ❑ Fonctions de hachage
- ❑ Les générateurs d'empreintes.
- ❑ Les générateurs de nombres aléatoires ou pseudo aléatoires.

Ces outils peuvent être utilisés seuls ou combinés pour réaliser des opérations de : Chiffrement, Scellement et signature, Échange de clés, Authentification mutuelle, etc.

Algorithmes de chiffrement symétrique

- ❖ Le principe d'utilisation d'une méthode (algorithme) de chiffrement symétrique est le suivant:
 - ✓ Un algorithme symétrique ou à clé secrète n'utilise qu'une seule clé
 - Cette clé ne doit être connue que des correspondants.
 - Cette clé sert aussi bien au chiffrement qu'au déchiffrement ($K_E = K_D = K$)
 - ✓ un message en clair est chiffré (crypté) en utilisant une méthode cryptographique avant d'être transmis au correspondant.
 - Le message chiffré peut être intercepté par un tiers sans danger, il est incompréhensible.
 - ✓ L'algorithme n'a pas besoin d'être secret.
 - Toute la sécurité repose sur le maintien secret d'une clé qui constitue l'élément variable dans le processus de chiffrement.
- ❖ Les méthodes à clés secrètes se divisent en deux types :
 - ✓ Les méthodes codant un flux de bits ou flots
 - ✓ Les méthodes de codage de blocs

Algorithmes de chiffrement symétrique

❑ chiffrement par blocs

L'idée générale du chiffrement par blocs est la suivante :

1. Remplacer les caractères par un code binaire
2. Découper cette chaîne en blocs de longueur donnée
3. Chiffrer un bloc en l'"additionnant" bit par bit à une clef.
4. Déplacer certains bits du bloc.
5. Recommencer éventuellement un certain nombre de fois l'opération 3.
6. Passer au bloc suivant et retourner au point 3 jusqu'à ce que tout le message soit chiffré.

➤ On distingue trois catégories de chiffrement par bloc :

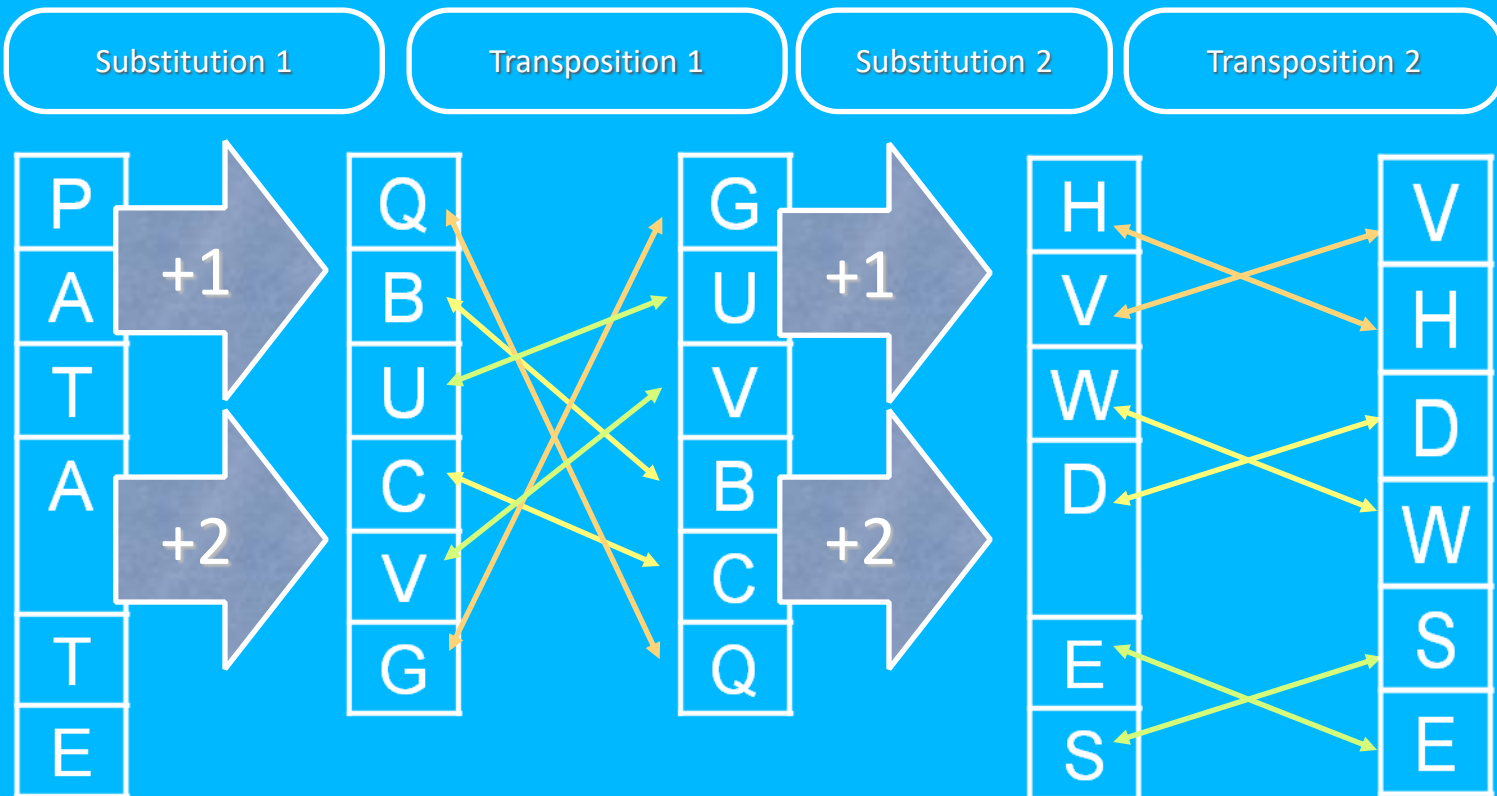
- ✓ Chiffrement par substitution : Les substitutions consistent à remplacer des symboles ou des groupes de symboles par d'autres symboles ou groupes de symboles dans le but de créer de la confusion.
- ✓ Chiffrement par transposition : Les transpositions consistent à mélanger les symboles ou les groupes de symboles d'un message clair suivant des règles prédéfinies pour créer de la confusion. Ces règles sont déterminées par la clé de chiffrement.
- ✓ Chiffrement par produit : C'est la combinaison des deux.

Exemples: AES, 3DES, Blowfish, IDEA, Cast5, etc.,

Algorithmes de chiffrement symétrique

❑ chiffrement par blocs

Un exemple commun est l'utilisation de transpositions (permutations), de substitutions et d'opérations linéaires simples.



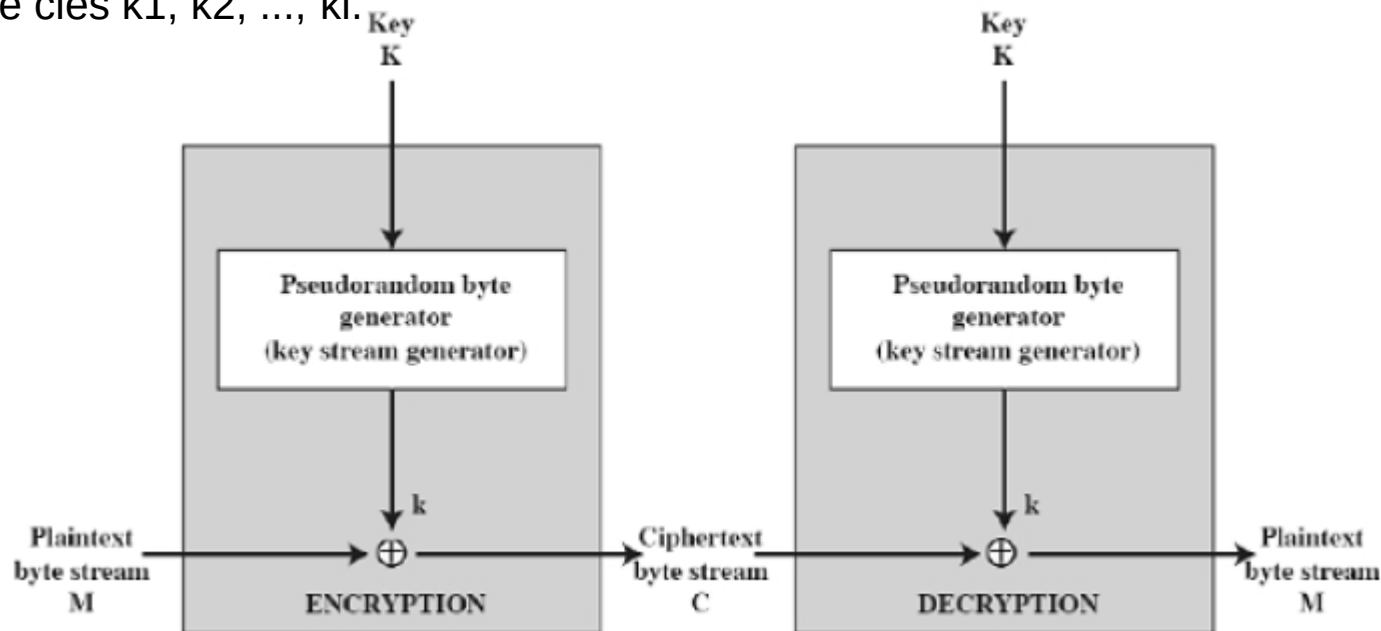
Algorithmes de chiffrement symétrique

❑ chiffrement par flots

L'opération de chiffrement s'opère sur chaque élément du texte clair (caractère, bits).

➤ On chiffre un bit/caractère à la fois.

➤ La structure d'un chiffrement par stream repose sur un générateur de clés qui produit une séquence de clés $k_1, k_2, \dots, k_i$.



➤ RC4 (utilisé dans SSL et dans le WiFi 802.11), E0/1 (norme Bluetooth), A5/1, A5/2, A5/3 (utilisés dans le GSM), Helix, SEAL, Rabbit

Algorithmes de chiffrement symétrique

❑ Avantages et inconvénients

- ✓ L'avantage principal de ce mode de chiffrement est sa rapidité
- ✓ Le principal désavantage réside dans la distribution des clés :
 - pour une meilleure sécurité, on pratiquera à l'échange de manière manuelle.
Malheureusement, pour de grands systèmes, le nombre de clés peut devenir conséquent.
 - En effet, pour un système à N utilisateurs, il y aura $N.(N - 1)/2$ paires de clés

❑ Quelques algorithmes de chiffrement symétrique et leur taille de clé

Taille des clés: Les algorithmes à clés secrètes utilisent des clés plus ou moins longues.

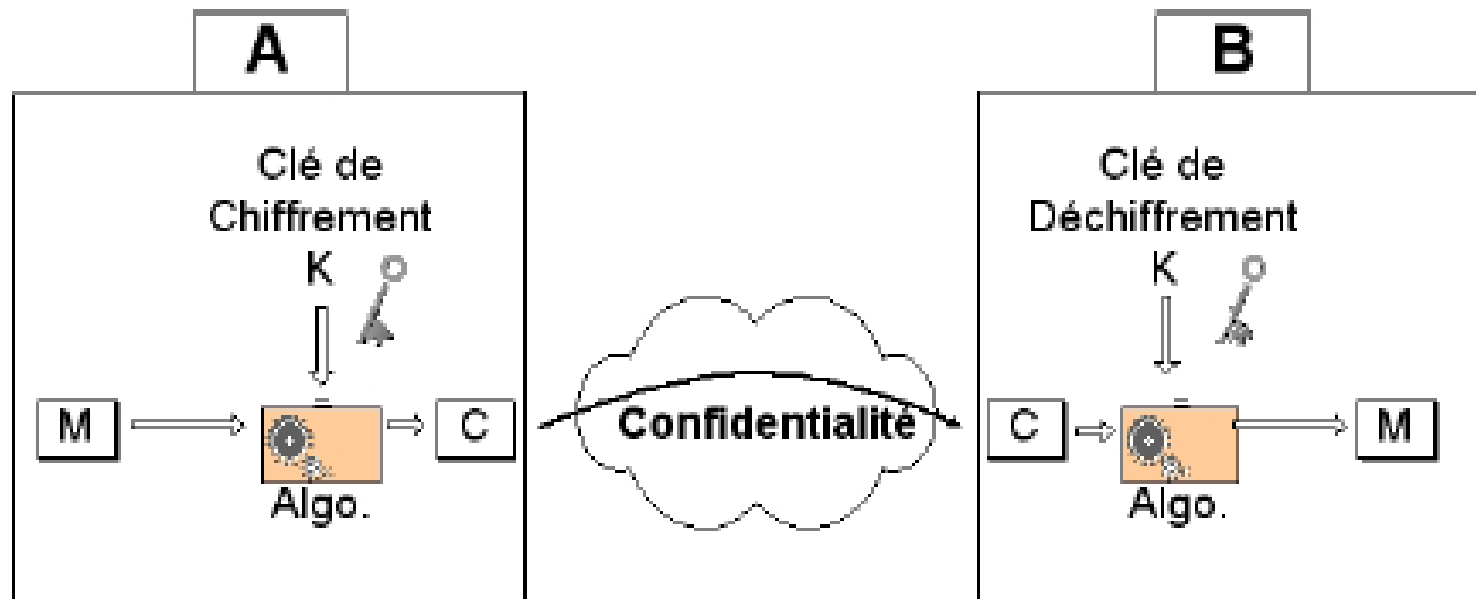
AES	Utilise des clés de 128, 192 ou 256 bits.
Camellia	Utilise des clés de 128, 192 ou 256 bits.
3DES	Utilise des clés de 112 bits.
DES	Utilise des clés de 56 bits.
RC4	Utilise des clés de 40 à 256 bits (chiffrement par flux)
RC5	Utilise des clés de 40 à 2040 bits (chiffrement par blocs)
RC6	Utilise des clés de 128 à 2040 bits (chiffrement par blocs)
Blowfish	Utilise des clés de 32 à 448 bits.
CAST5	Utilise des clés de 128 bits.
IDEA	Utilise des clés de 128 bits.

Algorithmes de chiffrement symétrique

❑ Service de Confidentialité d'un système symétrique

Elle est réalisée par le chiffrement du message.

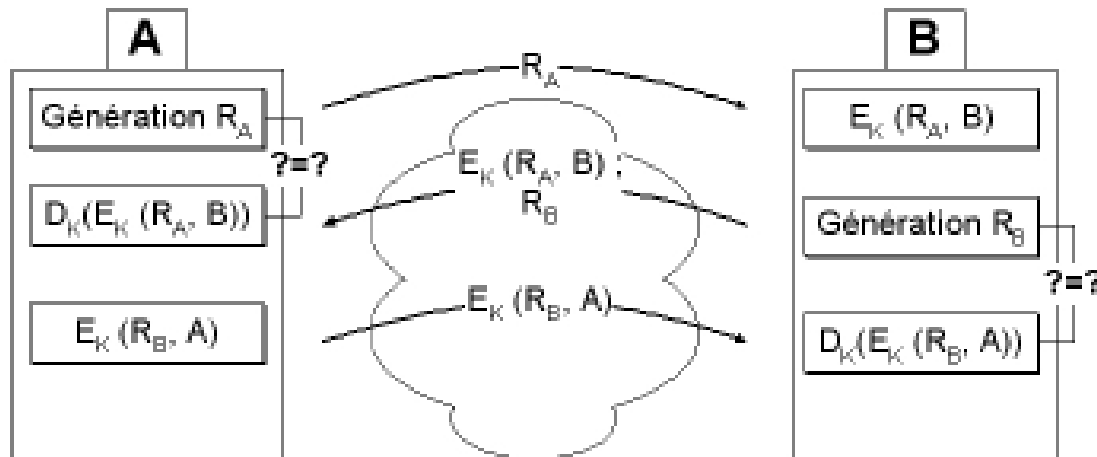
- ✓ Ici, la même clé est utilisée pour le chiffrement $E(M, K)$ et pour déchiffrement $D(C, K)$.
- ✓ Ce type de chiffrement nécessite un échange sûr préalable de la clé K entre les entités A et B .



Algorithmes de chiffrement symétrique

❑ Service d'Authentification dans un système symétrique

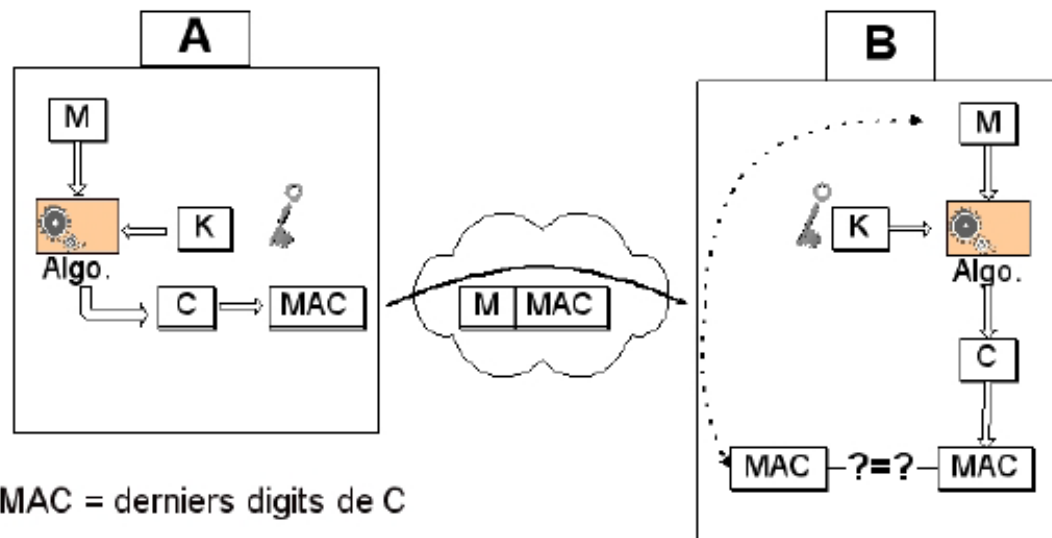
Un nonce R_A (p. ex. nombre aléatoire), propre à l'utilisateur A est envoyée. Les lettres A et B représentent des identificateurs personnels



Algorithmes de chiffrement symétrique

❑ Service d'Authentification message par MAC et système symétrique (scellement)

- Par l'utilisation d'un MAC (Message Authentication Code) généré à l'aide d'un cryptosystème à clé symétrique où le MAC est constitué des derniers chiffres de C, la clé secrète K utilisée étant partagée par les deux entités A et B. L'authentification repose sur l'utilisation de la clé K.



MAC = petit bloc authenticateur de taille fixe créé par un algorithme symétrique en se basant le message à envoyer et la clé secrète. Le dernier bloc du cryptogramme obtenu avec un algorithme de chiffrement symétrique en mode CBC: HMAC (HMAC-MD5, HMAC-SHA-1) ¹⁶

Algorithmes de chiffrement à clé publique

- ❑ Un algorithme asymétrique, dit également à clé publique, utilise deux clés, une clé publique et une clé privée.
 - Ces deux clés sont générées ensemble par un logiciel et elles dépendent mathématiquement l'une de l'autre.
 - Comme son nom l'indique, la clé publique peut être publiée sans risque, mais la clé privée doit être conservée jalousement secrète par son propriétaire.
 - La clé publique sert habituellement à crypter un message et la clé privée à le décrypter, mais l'inverse est possible.
 - La connaissance de la clé publique ne permet pas de déduire la clé privée.
 - La taille des clés s'étend de 512 bits à 2048 bits en standard.
 - Les algorithmes se basent sur des concepts mathématiques tels que l'exponentiation de grands nombres premiers, le problème des logarithmes discrets, ou encore le problème du sac à dos

Algorithmes de chiffrement à clé publique

❑ Avantages et inconvénients

- ✓ La distribution des clés est grandement facilitée car l'échange de clés secrètes n'est plus nécessaire.
 - Chaque utilisateur conserve sa clé secrète sans jamais la divulguer. Seule la clé publique devra être distribuée.
- ✓ A l'inverse du chiffrement symétrique où le nombre de clés est le problème majeur, ici, seules n paires sont nécessaires.
 - En effet, chaque utilisateur possède une paire (K_p, K_s) et tous les transferts de message ont lieu avec ces clés.
- ✓ Comme désavantage, le chiffrement par voie asymétrique est environ 100 à 1000 fois plus lent que le chiffrement symétrique et d'une taille de clés importante, au moins 2048 bits pour le RSA, pour un niveau de sécurité correct? Cela peut aboutir à des problèmes de stockage.

❑ Quelques algorithmes à clés publiques

- Diffie-Hellman.
- RSA (Ronald Rivest, Adi Shamir et Leonard Adleman).
- DSA ou DSS (Digital Signature Algorithm ou Digital Signature Standard) est un standard américain du NIST (National Institute of Standards and Technologies).
- Les méthodes elliptiques.

Algorithmes de chiffrement à clé publique

❑ Cas du RSA

On possède une paire de clés, l'une publique (e,n) et une privée (d,n) . La première étape revient à choisir n . Il doit s'agir d'une valeur assez élevée, produit de 2 nombres premiers très grands p et q . En pratique, si p et q ont 100 chiffres décimaux, n possèdera 200 chiffres. Selon le niveau de sécurité souhaité, la taille de n peut varier : 512 bits, 768, 1024 ou 2048.

Dans un second temps, on choisira un très grand entier e , relativement premier à $(p-1)*(q-1)$. La clé publique sera formée par (e,n) . On choisira ensuite un d tel que $e * d \equiv 1 \mod (\Phi(n))$.

La clé privée sera donnée par (d,n) .

Dernière phase : on jette p et q . Le cryptanalyste devant retrouver ces valeurs, il faut les détruire pour éviter les fuites.

Algorithmes de chiffrement à clé publique

❑ Cas du RSA

m = message en clair

c = message encrypté

(e,n) constitue la clé publique

(d,n) constitue la clé privée

n est le produit de 2 nombres premiers

mod est l'opération de modulo (reste de la division entière)

Pour créer une paire de clés

1. Prendre deux nombres premiers **p** et **q** (de taille à peu près égale). Dans la pratique ce sont de grands nombres premiers *réellement* aléatoires
2. Calculer **n = pq**.
3. Prendre un nombre **e** qui n'a aucun facteur en commun avec **(p-1)(q-1)**.
4. Calculer **d** tel que **ed mod (p-1)(q-1) = 1**
5. Le couple **(e,n)** constitue la clé publique. **(d,n)** est la clé privée.

Pour encrypter un message

On fait: **c = m^e mod n**

Pour décrypter:

m = c^d mod n

Algorithmes de chiffrement à clé publique

❑ Cas du DIFFIE HELLMAN

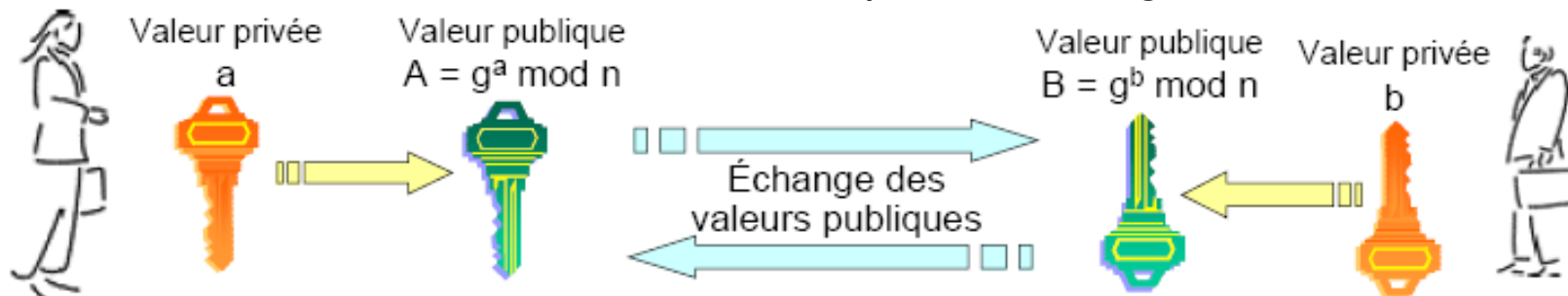
■ Qu'est-ce que le protocole DH ?

- ◆ Protocole cryptographique qui permet à deux tiers de générer un secret partagé sans informations préalables l'un sur l'autre

■ Principe

- ◆ Échange de valeurs publiques

Étant donnés des entiers n, g, a avec n premier et $1 \leq g \leq n-1$



- ◆ Permettant de générer un secret partagé



- ◆ Un espion ne peut reconstituer le secret partagé à partir des valeurs publiques

Algorithmes de chiffrement à clé publique

❑ Cas du DIFFIE HELLMAN

	Alice	Bob
Etape 1	Alice et Bob choisissent ensemble un grand nombre premier p et un entier g tel que $1 \leq g \leq n-1$. Cet échange n'a pas besoin d'être sécurisé	
Etape 2	Alice choisit secrètement a	Bob choisit secrètement b
Etape 3	Alice calcule $A = g^a \bmod n$	Bob calcule $B = g^b \bmod n$
Etape 4	Alice et Bob s'échangent les valeurs de A et B . Cet échange n'a pas besoin d'être sécurisé.	
Etape 5	Alice calcule la clé secrète $K = B^a \bmod n$	Bob calcule la clé secrète $K = A^b \bmod n$

- La sécurité de la méthode repose sur la difficulté de calcul de $k = g^{ab} \bmod n$ à partir de $g^a \bmod n$ et de $g^b \bmod n$ lorsque n est grand.
- La fonction $f(x) = g^x \bmod n$ est une fonction à sens unique : $f(x)$ est aisé à calculer, mais retrouver x à partir de $f(x)$, g et n est très délicat (si $y = g^x \bmod n$, on dit que x est le logarithme discret de y de base g modulo n). Ce problème mathématique est connu sous le nom du problème du logarithme discret

Algorithmes de chiffrement à clé publique

❑ Cas du DIFFIE HELLMAN

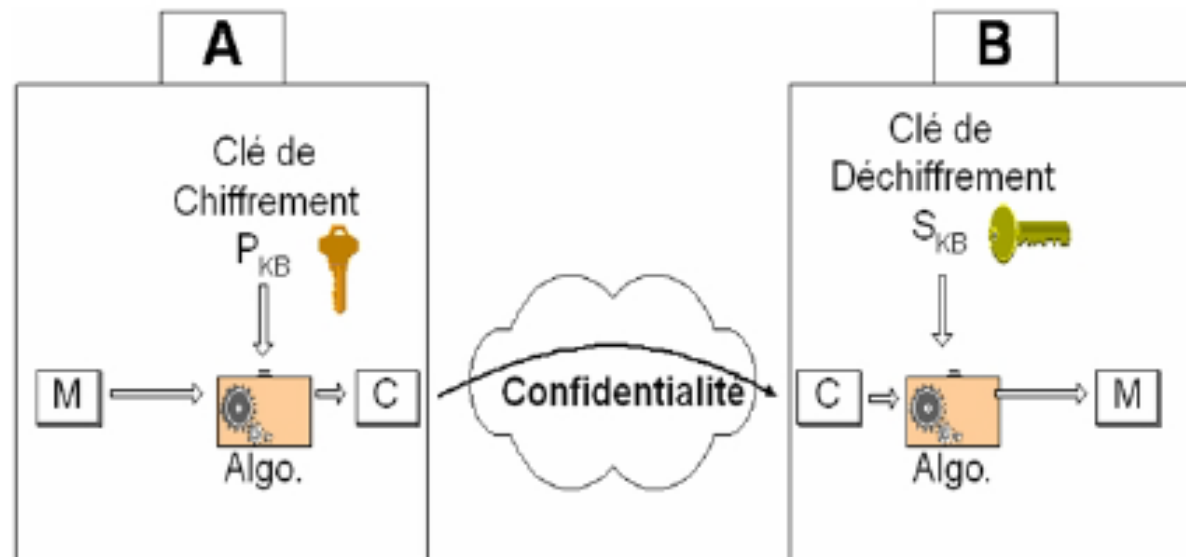
Le procédé de Diffie-Hellman présente deux inconvénients majeurs:

- Quand Alice veut transmettre un message chiffré à Bob, elle doit d'abord entrer en contact avec lui afin d'effectuer un premier échange ; chaque transmission de message s'en trouve ainsi fortement ralentie et perd toute instantanéité (sans parler de l'augmentation du nombre de communications engendrée par le système).
- Une tierce personne peut intervenir lors du processus d'échange des valeurs publiques en les remplaçant par des valeurs de son choix ; elle sera alors en mesure de déchiffrer les messages transmis sans que les correspondants ne s'en aperçoivent. Par exemple, lors de l'échange des valeurs,
 - ✓ l'intermédiaire intercepte g, n, A et B . Il choisit sa propre valeur privée c et calcule $C = g^c \bmod n$ qui donne à Alice et à Bob. Il calcule $k_1 = A^c \bmod n$ et $k_2 = B^c \bmod n$
 - ✓ Alice va utiliser la clé $k_1 = g^{ac} \bmod n$ et Bob la clé $k_2 = g^{bc} \bmod n$.
 - ✓ Lors de l'échange d'un message chiffré, l'intermédiaire déchiffre les messages d'Alice à l'aide de la clé $k_1 = g^{ac} \bmod n$ qu'il partage avec Alice, puis les chiffre avec la clé $k_2 = g^{bc} \bmod n$ qu'il partage avec Bob. Ce dernier va déchiffrer le message d'Alice sans s'apercevoir qu'il a été intercepté et manipulé.

Algorithmes de chiffrement à clé publique

❑ Service de Confidentialité d'un système asymétrique

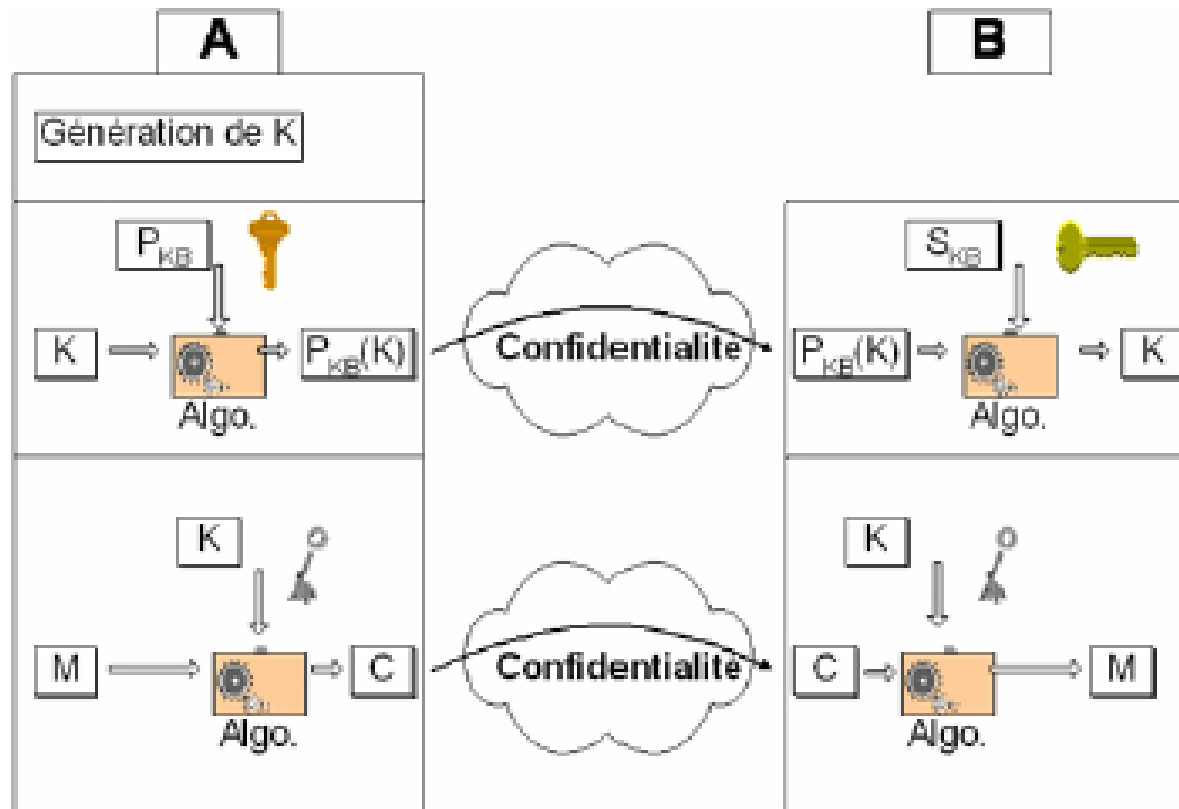
Ici, chaque entité possède sa propre paire de clés. On aura donc la paire P_{KA} , S_{KA} pour l'entité A et la paire P_{KB} , S_{KB} pour l'entité B.



Algorithmes de chiffrement à clé publique

❑ Service de Confidentialité d'un système asymétrique

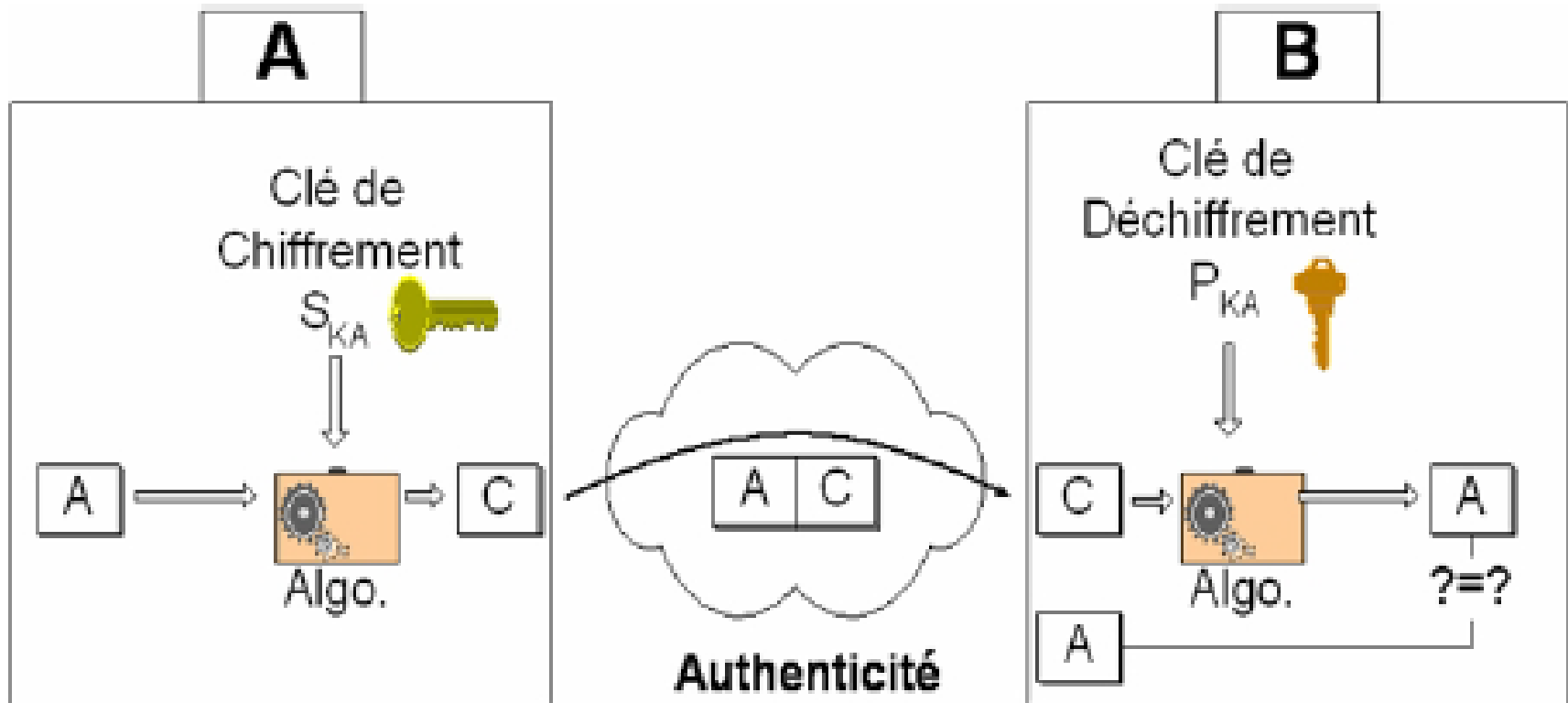
Par l'intermédiaire du système à clé publique, on sécurise l'échange de la clé K du système symétrique. Ensuite, les deux parties ayant acquis de manière sécurisée cette clé de chiffrement K , on utilisera le système à clé symétrique pour chiffrer le message



Algorithmes de chiffrement à clé publique

❑ Service d'Authentification dans un système asymétrique

Comme le propriétaire de cette clé est le seul à la connaître, cela prouve qu'il est bien la personne ayant chiffré le message.



Role et Principe

- ❑ Les fonctions de hachage (ou générateurs d'empreintes ou « hash functions ») ont un rôle similaire à celui des calculs de « CRC »:
 - ✓ permettent de créer l'empreinte numérique d'un message. Cette empreinte dépend mathématiquement du message.
 - Si l'on modifie un tant soit peu le message, l'empreinte associée est complètement différente.
 - Par conséquent les fonctions de hachage vérifient qu'un message (ou un fichier) n'a pas été altéré.
- ❑ Le principe est qu'un message clair de longueur quelconque doit être transformé en un message de longueur fixe inférieure à celle de départ
 - Le message réduit portera le nom de "Haché" ou de "Condensat".
 - L'intérêt est d'utiliser ce condensat comme empreinte digitale du message original afin que ce dernier soit identifié de manière presque univoque.

NB: En clair, des messages différents peuvent avoir les mêmes empreintes.

- ✓ Exemples : Les principaux algorithmes de hachage sont : MD5, SHA, MD2, MD4, N-Hash, Snefrou, Ripend160, MDC2...

Exemples

❑ Quelques fonctions de hachage

- MD2,
- MD4,
- MD5,
- SHA,
- N-Hash,
- Snefrou,
- Ripend160,
- MDC2...

Conseils

Du fait de l'augmentation de la puissance des ordinateurs domestiques (ou plutôt de leur carte graphique), les mots de passe doivent être protégés par des algorithmes qui demandent beaucoup de temps de calcul. Certains systèmes récents ont abandonné MD5 pour SHA-512 pour ces raisons.

Exemples

Administrateur : Invite de commandes

— □

```
CertUtil: -? La commande s'est terminée correctement.
```

```
C:\WINDOWS\system32>certUtil -hashfile /?
```

Utilisation :

```
CertUtil [Options] -hashfile FichierEntrée [AlgorithmeHachage]
Générer et afficher le hachage de chiffrement sur
un fichier
```

Options :

```
-Unicode          -- Écrire la sortie redirigée au format Unicode
-gmt              -- Afficher les heures GMT
-seconds          -- Afficher le temps en secondes et millisecondes
-v               -- Opération en mode détaillé
-privatekey       -- Afficher les données de mot de passe et de clé privée
-pin Code PIN     -- Code PIN de la carte à puce
-sid WELL_KNOWN_SID_TYPE -- SID numérique
    22 -- Système local
    23 -- Service local
    24 -- Service réseau
```

```
Algorithmes de hachage : MD2 MD4 MD5 SHA1 SHA256 SHA384 SHA512
```

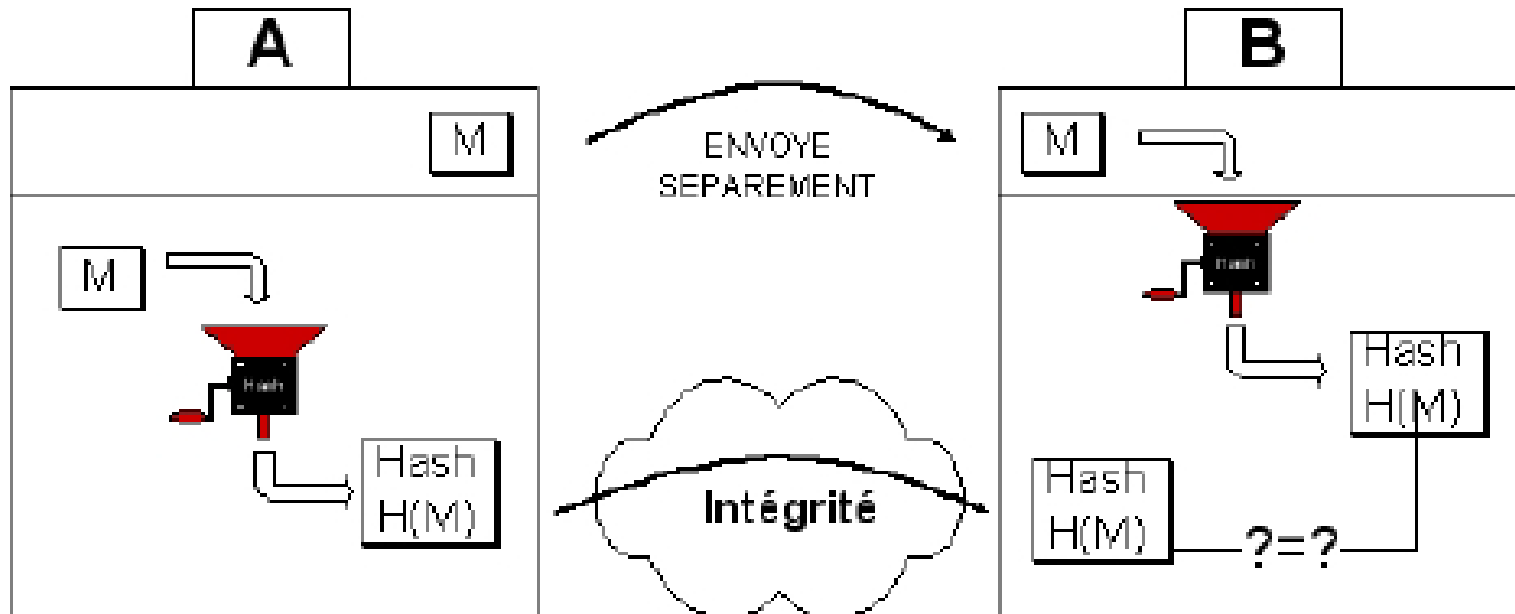
```
CertUtil -?      -- Afficher une liste de verbes (liste de commandes)
```

```
CertUtil -hashfile -? -- Afficher le texte d'aide pour le verbe "hashfile"
```

Cas d'utilisation

❑ Vérification de l'intégrité par fonction de hachage

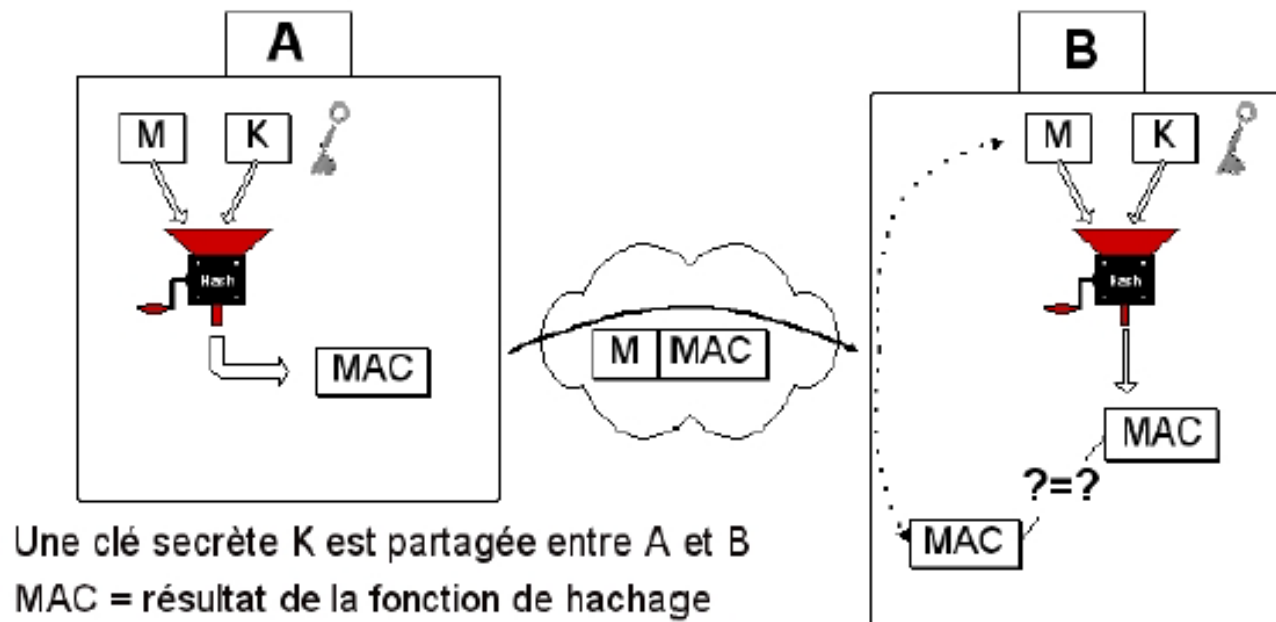
Ici, chaque entité possède sa propre paire de clés. On aura donc la paire P_{KA} , S_{KA} pour l'entité A et la paire P_{KB} , S_{KB} pour l'entité B.



Cas d'utilisation

❑ Authentification message par MAC et système symétrique (scellement): HMAC

- Par l'utilisation d'un MAC (Message Authentication Code) généré à l'aide d'une fonction de hachage et de la clé secrète K partagée par les deux entités A et B. C'est l'équivalent de la signature électronique pour la cryptographie symétrique



Ce procédé est très rapide car ne faisant intervenir aucun algorithme de chiffrement. Il est donc adapté à la protection de trafic temps-réel.

Cas d'utilisation

❑ Authentification par signature (technique asymétrique)

Par l'utilisation d'une signature digitale. Parmi les propriétés remarquables de ces signatures, on peut dire qu'elles doivent être authentiques, infalsifiables, non-réutilisables, non-répudiables, et inaltérables.

